



11

**\*\*FILE\*\*ID\*\*DXUTILITY**

(1) 72 RX FUNCTION DECISION TABLE  
(1) 131 START I/O OPERATION

```
0000 1 .TITLE DXUTILITY - FLOPPY DISK DRIVER UTILITY ROUTINES
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 ****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *
0000 26 ****
0000 27
0000 28 C. A. MONIA 25-JUN-77
0000 29
0000 30 MODIFIED BY:
0000 31
0000 32 V03-001 PRD0021 Paul R. DeStefano 27-Apr-1983
0000 33 Corrected computation of next media address to accept sector
0000 34 addresses in the range of 1 to 26 when doing physical I/O.
0000 35
0000 36 V02-003 TCM0001 Trudy C. Matthews 19-Jun-1981
0000 37 Clear R1 (which becomes the second longword of the IOSB)
0000 38 before exiting. This is because certain code paths through
0000 39 I/O completion (notably for paging I/O and swapping I/O,
0000 40 when this driver is being used for standalone sysgen systems)
0000 41 expect the second longword of the IOSB to be zero.
0000 42
0000 43
0000 44 STAR FLOPPY DRIVER UTILITY ROUTINES
0000 45
0000 46 MACRO LIBRARY CALLS
0000 47
0000 48
0000 49 $ADPDEF :DEFINE ADP OFFSETS
0000 50 $CRBDEF :DEFINE CRB OFFSETS
0000 51 $EMBDEF :DEFINE EMB OFFSETS
0000 52 $IDBDEF :DEFINE IDB OFFSETS
0000 53 $IODEF :DEFINE I/O FUNCTION CODES
0000 54 $IRPDEF :DEFINE IRP OFFSETS
0000 55 $PRDEF :DEFINE PROCESSOR REGISTERS
0000 56 $SSDEF :DEFINE SYSTEM STATUS VALUES
0000 57 $UBADEF :DEFINE UBA REGISTER OFFSETS
```

```
0000 58      $UCBDEF          ;DEFINE UCB OFFSETS
0000 59      $VECDEF          ;DEFINE INTERRUPT DISPATCH VECTOR OFFSETS
0000 60
0000 61
0000 62      : LOCAL MACROS
0000 63
0000 64      : REPORT DEVICE ERROR
0000 65
0000 66
0000 67      .MACRO RPTERR CODE
0000 68      MOVZWL CODE, R0
0000 69      BRB    FUNCXT
0000 70      .ENDM
```

```

0000  72      .SBTTL RX FUNCTION DECISION TABLE
0000  73 :+
0000  74 : RX FUNCTION DECISION TABLE
0000  75 :-
0000  76
00000000 77      .PSECT $SS115_DRIVER.LONG
0000  78  DX$FUNCTIONTABLE::: ;FUNCTION DECISION TABLE
0000  79  FUNCTAB :- ;LEGAL FUNCTIONS
0000  80  <SENSECHAR,- ;SENSE CHARACTERISTICS
0000  81  SETCHAR,- ;SET CHARACTERISTICS
0000  82  SENSEMODE,- ;SENSE MODE
0000  83  SETMODE,- ;SET MODE
0000  84  READLBLK,- ;READ LOGICAL BLOCK
0000  85  WRITELBLK,- ;WRITE LOGICAL BLOCK
0000  86  READPBLK,- ;READ PHYSICAL BLOCK
0000  87  WRITEPBLK,- ;WRITE PHYSICAL BLOCK
0000  88  READVBLK,- ;READ VIRTUAL BLOCK
0000  89  WRITEVBLK,- ;WRITE VIRTUAL BLOCK
0000  90  ACCESS,- ;ACCESS FILE AND/OR FIND DIRECTORY ENTRY
0000  91  ACPCONTROL,- ;ACP CONTROL FUNCTION
0000  92  CREATE,- ;CREATE FILE AND/OR DIRECTORY ENTRY
0000  93  DEACCESS,- ;DEACCESS FILE
0000  94  DELETE,- ;DELETE FILE AND/OR DIRECTORY ENTRY
0000  95  MODIFY,- ;MODIFY FILE ATTRIBUTES
0000  96  MOUNT> ;MOUNT VOLUME
0008  97  FUNCTAB :- ;BUFFERED I/O FUNCTIONS
0008  98  <SENSECHAR,- ;SENSE CHARACTERISTICS
0008  99  SETCHAR,- ;SET CHARACTERISTICS
0008 100  SENSEMODE,- ;SENSE MODE
0008 101  SETMODE,- ;SET MODE
0008 102  ACCESS,- ;ACCESS FILE AND/OR FIND DIRECTORY ENTRY
0008 103  ACPCONTROL,- ;ACP CONTROL FUNCTION
0008 104  CREATE,- ;CREATE FILE AND/OR DIRECTORY ENTRY
0008 105  DEACCESS,- ;DEACCESS FILE
0008 106  DELETE,- ;DELETE FILE AND/OR DIRECTORY ENTRY
0008 107  MODIFY,- ;MODIFY FILE ATTRIBUTES
0008 108  MOUNT> ;MOUNT VOLUME
0010 109  FUNCTAB +ACPSREADBLK,- ;READ FUNCTIONS
0010 110  <READLBLK,- ;READ LOGICAL BLOCK
0010 111  READPBLK,- ;READ PHYSICAL BLOCK
0010 112  READVBLK> ;READ VIRTUAL BLOCK
001C 113  FUNCTAB +ACPSWRITEBLK,- ;WRITE FUNCTIONS
001C 114  <WRITELBLK,- ;WRITE LOGICAL BLOCK
001C 115  WRITEPBLK,- ;WRITE PHYSICAL BLOCK
001C 116  WRITEVBLK> ;WRITE VIRTUAL BLOCK
0028 117  FUNCTAB +ACPSACCESS,<ACCESS,CREATE> ;ACCESS AND CREATE FILE OR DIRECTORY
0034 118  FUNCTAB +ACPSDEACCESS,<DEACCESS> ;DEACCESS FILE
0040 119  FUNCTAB +ACPSMODIFY,- ;MODIFY
0040 120  <ACPCONTROL,- ;ACP CONTROL FUNCTION
0040 121  DELETE,- ;DELETE FILE OR DIRECTORY ENTRY
0040 122  MODIFY> ;MODIFY FILE ATTRIBUTES
004C 123  FUNCTAB +ACPSMOUNT,<MOUNT> ;MOUNT VOLUME
0058 124  FUNCTAB +EXESSENSEMODE,- ;SENSE CHARACTERISTICS
0058 125  <SENSECHAR,- ;SENSE MODE
0058 126  SENSEMODE> ;SENSE MODE
0064 127  FUNCTAB +EXESSETCHAR,- ;SET CHARACTERISTICS
0064 128

```

DXUTILITY  
V04-000

E 12  
- FLOPPY DISK DRIVER UTILITY ROUTINES      15-SEP-1984 23:57:49 VAX/VMS Macro V04-00  
RX FUNCTION DECISION TABLE      5-SEP-1984 00:14:19 [DRIVER.SRC]DXUTILITY.MAR;1 Page 4 (1)

0064 129

SETMODE>

;SET MODE

50	00	21	2A	A3	08	E0	0070
38	A3	38	A3	04	04	7A	0075
A3	50	52	44	A5	9A	0078	181
51	50	52	52	45	A5	7B	007F
39	A3	52	50	52	9A	0085	182
	51	39	A3	51	90	0089	183
						184	183
						185	184
						186	185
						187	186

0070 131 .SBTTL START I/O OPERATION  
 0070 132  
 0070 133 :+  
 0070 134 DX\$STARTIO - START I/O ON FLOPPY DEVICE UNIT  
 0070 135 THIS ROUTINE IS ENTERED VIA A 'BSBW' TO START I/O ON A DEVICE UNIT.  
 0070 136 CONTROL ALTERNATES BETWEEN THE FLOPPY DRIVER AND THIS CODE. THIS ROU-  
 0070 137 TINE IS CALLED TO PERFORM HARDWARE INDEPENDANT PROCESSING. ALL HARD-  
 0070 138 WARE SPECIFIC PROCESSING IS PERORMED BY DEVICE-SPECIFIC CODE IN THE  
 0070 139 DRIVER.  
 0070 140  
 0070 141  
 0070 142  
 0070 143  
 0070 144  
 0070 145  
 0070 146  
 0070 147  
 0070 148  
 0070 149  
 0070 150  
 0070 151  
 0070 152  
 0070 153  
 0070 154  
 0070 155  
 0070 156  
 0070 157  
 0070 158  
 0070 159  
 0070 160  
 0070 161  
 0070 162 EACH CO-ROUTINE CALL SITE CONTAINS A CONTINGENCY EXIT ADDRESS. IN THE  
 0070 163 EVENT OF A HARDWARE ERROR, CONTROL WILL BE PASSED TO THE ERROR HANDLER  
 0070 164 TO EFFECT A RE-TRY OR TERMINATE THE REQUEST.  
 0070 165  
 0070 166 INPUTS:  
 0070 167  
 0070 168 R3 = ADDRESS OF I/O PACKET  
 0070 169  
 0070 170 R5 = UCB ADDRESS OF DEVICE UNIT  
 0070 171  
 0070 172 OUTPUTS:  
 0070 173  
 0070 174 \*\*\*\*\*OUTPUTS\*\*\*\*\*  
 0070 175  
 0070 176 :-  
 0070 177  
 0070 178 .ENABL LSB  
 0070 179  
 0070 180 DX\$STARTIO:: :START I/O OPERATION  
 BBS #IRPSV PHYSIO,IRPSW STS(R3),1\$ ;BYPASS BLOCK COMPUTATION IF PHYSICAL  
 EMUL #4,IRPSL MEDIA(R3),R0,R0 ;SCALE LBN, QUAD RESULT TO R0  
 MOVZBL UCBSL DEVDEPEND(R5),R2 ;GET NUMBER OF SECTORS PER TRACK  
 EDIV R2,R0,R0,IRPSL MEDIA(R3) ;COMPUTE SECTOR  
 MOVZBL UCBSL DEVDEPEND+1(R5),R2 ;GET TRACKS PER CYLINDER  
 EDIV R2,R0,R0,R1 ;COMPUTE TRACKS (R1), CYL. (R0)  
 MOV8 R1,IRPSL MEDIA+1(R3) ;SAVE TRACK ADDRESS

3A A3 50	B0 0092	188	MOVW R0,IRPSL_MEDIA+2(R3)	;SAVE CYLINDER ADDRESS
00D8 C5 7E A5 00 68 A5 03	B0 0096	189	1\$: MOVW UCBSW_BCNT(R5),UCBSW_DX_BCR(R5)	;STORE BYTE COUNT
	E5 009C	190	BBCC #UCBSV_DX_WRITE,UCBSB_DEVSTS(R5),2\$	;CLEAR WRITE FLAG
	00A1	191		
50 20 A3 06 00 OC 50 08	0126 30	00A1 193	2\$: BSBW SETBUF	;SETUP BUFFER PARAMETERS
	EF 00A4	194	EXTZV #IRPSV_FCODE, #IRPSS_FCODE, IRPSW_FUNC(R3), R0	;GET FUNCTION CODE
	91 00AA	195	CMPB R0, #10\$_READPBLK	;READ PHYSICAL BLOCK?
	13 00AD	196	BGEQ 5\$	;IF EQL YES
00 68 A5 03	E2 00AF	197	BBSS #UCBSV_DX_WRITE,UCBSW_DEVSTS(R5),3\$	;SET WRITE FLAG
	00EE 30	00B4 198		
	00B7	199	BSBW MOVFRUSER	;GET INITIAL SECTOR FULL OF DATA
0080 C5 0081 C5 04 20 A3 0F	90 00B7	201	5\$: MOVB UCBSB_ERTRMAX(R5),UCBSB_ERTCNT(R5)	;INITIALIZE ERROR RETRY COUNT
	E1 00BE	202	BBC #10\$V_INHRETRY, IRPSW_FONC(R3), 10\$	;IF BIT CLEAR, PERFORM NORMAL RETR
	94 00C3	203	CLRB UCBSB_ERTCNT(R5)	;INHIBIT RETRIES ON ERROR
46 A5 02 A2 46	0081 30	00C7 205	10\$: BSBW TRKSEC	;COMPUTE MEDIA ADDRESS
	91 00CA	206	CMPB ?(R2),UCBSW_CYLINDERS(R5)	;LEGAL DISK ADDRESS?
	1E 00CF	207	BGEQU 70\$	;IF GEQU NO
64 A5 00E8 8F	AA 00D1	208	BICW #UCBSM_CANCEL!-	;CLEAR CANCEL I/O.
	00D7	209	UCBSM_INTTYPE!=	;INTERRUPT TYPE
	00D7	210	UCBSM_POWER!=	;POWERFAIL AND
	00D7	211	UCBSM_TIMEOUT,UCBSW_STS(R5)	;TIMEOUT STATUS BITS
	00D7	212		
53 D6 00D7 9E 16 00D9 53 D4 00DB 00D0 C5 D6 00DA C5 97 00E1 FO 12 00E5 9E 16 00E7 0095 30 00E9 3F 50 E9 00EC	213 214 215 216 217 218 219 220 221 222 223 224 225 226		20\$: INCL R3 JSB @(SP)+ CLRL R3 INCL UCBSL_DX_BFPNT(R5) DEC8 UCBSB_DX_SCTCNT(R5) BNEQ 20\$ JSB @(SP)+ BSBW XFER BLBC R0,IOSUCC	;FLAG READY FOR TRANSFER ;SEEK/TRANSFER ONE BYTE OF DATA ;ASSUME TRANSFERRED LAST BYTE ;INCREMENT BUFFER POINTER ;DECREMENT SECTOR COUNT ;IF NEQ TRANSFER ANOTHER BYTE ;CALL THE CALLER ;TRANSFER DATA TO/FROM USER ;IF LBC DONE
	00EF	223		
	00EF	224		
	00EF	225		
	00EF	226		
	00EF	227		
53 58 A5 00 38 A3 96 00F3 38 A3 91 00F6 44 A5 00F9 08 E0 00FB 07 2A A3 00FD C5 1F 0100 38 A3 94 0102 06 11 0105 BE 18 0107	227 228 229 230 231 232 233 234 235 236 237 50\$:		MOVL UCBSL_IRP(R5), R3 INC8 IRPSL_MEDIA(R3) CMPB IRPSL_MEDIA(R3) - UCBSB_SECTORS(R5) BBS #IRPSV_PHYSIO,- IRPSW_STS(R3),50\$ BLSSU 10\$	;Get address of I/O packet. ;Increment sector address. ;Is the new sector address greater ;than the number of sectors per track? ;Branch if performing physical I/O.
	0109	238	CLRB IRPSL_MEDIA(R3)	
	0100	239	BRB 60\$	
	0110	240	BLEQU 10\$	
38 A3 01 3A A3 96 3A A3 91 46 A5 0113 80 1F	0109 240 60\$: 0100 241 0110 242 0115 243 0117 244		MOV8 #1,IRPSL_MEDIA(R3) INC8 IRPSL_MEDIA+2(R3) CMPB IRPSL_MEDIA+2(R3) - UCBSW_CYLINDERS(R5) BLSSU 10\$	
			RPTERR #SSS_IVADDR	
		70\$:		

011E 245  
 011E 246  
 011E 247  
 011E 248 :+ DXSERR - CONTINGENCY EXIT HANDLER  
 011E 249  
 011E 250 THIS ROUTINE IS ENTERED WHENEVER THE DRIVER DETECTS A POWER-FAIL, DEVICE  
 011E 251 TIMEOUT OR HARDWARE ERROR CONDITION. IF THE ERROR WAS CAUSED BY DEVICE  
 011E 252 TIMEOUT OR POWER FAIL, THE TRANSFER IS RESTARTED. IF THE ERROR IS NON-  
 011E 253 FATAL, THE RETRY COUNT IS DECREMENTED AND THE TRANSFER IS REPEATED IF  
 011E 254 THE RESULT IS NONZERO. ALL OTHER CONDITIONS RESULT IN REQUEST TERMINA-  
 011E 255 TION WITH STATUS CONTAINED IN R0  
 011E 256  
 011E 257 INPUTS:  
 011E 258  
 011E 259 R0 = ERROR STATUS CODE  
 011E 260 R3 = ERROR SEVERITY INDICATION  
 011E 261  
 011E 262 R5 = ADDRESS OF UCB  
 011E 263 (SP) = RETURN TO DRIVER RESTART CODE  
 011E 264  
 011E 265 R3 LBC = FATAL ERROR  
 011E 266 R3 LBS = RETRIABLE ERROR  
 011E 267  
 011E 268 OUTPUTS:  
 011E 269  
 011E 270 NONE  
 011E 271 :-  
 011E 272  
 011E 273 DXSERR::  
 A1 64 10 53 E9 011E 274 BLBC R3,FUNCXT :IF LBC, FATAL HARDWARE ERROR  
 A5 05 E0 0121 275 BBS #UCBSV\_POWER,UCBSW\_STS(R5),10\$ :RETRY ON POWER FAIL  
 0080 C5 97 0126 276 DECB UCBSB\_ERTCNT(R5) :DECREMENT RETRY COUNT  
 98 14 012A 277 BGTR 10\$ :IF GTR, TRY AGAIN  
 03 11 012C 278 BRB FUNCXT :TERMINATE REQUEST  
 012E 279  
 012E 280 .DSABL LSB  
 012E 281  
 012E 282  
 012E 283 : TERMINATE REQUEST SUCCESSFULLY  
 012E 284  
 012E 285  
 50 01 3C 012E 286 IOSUCC: MOVZWL #SSS\_NORMAL,R0 :SET SUCCESS  
 0131 287  
 0131 288  
 0131 289  
 0131 290 : FUNCTION COMPLETION COMMON EXIT  
 0131 291  
 0131 292  
 0131 293 FUNCXT:  
 02 AE 7E A5 00000000 GF 0131 294 CLRL (SP) :ZERO COROUTINE ADDRESS ON STACK  
 50 00DB C5 16 0133 295 PUSHL R0 :SAVE REGISTER  
 03 BA 0135 296 JSB G^IOC\$DIAGBUFILL :FILL DIAGNOSTIC BUFFER IF PRESENT  
 0143 297 SUBW3 UCBSW\_DX\_BCR(R5),UCBSW\_BCNTR5,2(SP) :COMPUTE BYTES TRANSFERRED  
 0145 298 POPR #^M<R0,RT> :RESTORE R0 AND PUT ZERO IN R1  
 0148 299 REQCOM :TERMINATE REQUEST  
 014B 300  
 014B 301 ;+

014B 302 : TRKSEC - CONVERT LOGICAL TO PHYSICAL DISK ADDRESS

014B 303

014B 304 : THIS ROUTINE IS ENTERED VIA A 'BSB' TO CONVERT A LOGICAL DISK

014B 305 : ADDRESS TO A PHYSICAL ADDRESS. IF THE PHYSICAL I/O FLAG IS SET

014B 306 : IN THE I/O REQUEST PACKET, THE CONVERSION CONSISTS OF SIMPLY

014B 307 : MOVING THE LOGICAL TRACK, SECTOR AND CYLINDER ADDRESSES IN

014B 308 : THE PACKET MEDIA LONGWORD TO THE MEDIA ADDRESS LONGWORD IN

014B 309 : THE UCB. IF LOGICAL I/O IS BEING PERFORMED, THEN THE LOGICAL

014B 310 : ADDRESS IN THE I/O PACKET IS CONVERTED TO A PHYSICAL ADDRESS

014B 311 : BY APPLYING INTERLEAVE AND TRACK-TO-TRACK SKEW. THE RESULT IS

014B 312 : PLACED IN THE UCB MEDIA ADDRESS LONGWORD.

014B 313

014B 314 : INPUTS:

014B 315

014B 316

014B 317 : R5 = ADDRESS OF UCB

014B 318

014B 319 : OUTPUTS:

014B 320

014B 321 : R2 = POINTER TO PHYSICAL MEDIA ADDRESS

014B 322

014B 323 : UCBSL\_MEDIA CONTAINS THE PHYSICAL MEDIA ADDRESS

014B 324

014B 325

014B 326 : -

014B 327

014B 328 : TRKSEC:

53	58	A5	D0	014B 329 : MOVL UCBSL_IRP(R5), R3 : GET ADDRESS OF REQUEST PACKET
52	00BC	C5	9E	014F 330 : MOVAB UCBSL_MEDIA(R5), R2 : POINT TO PHYSICAL MEDIA ADDRESS
62	38	A3	D0	0154 331 : MOVL IRPSL_MEDIA(R3), (R2) : COPY LOGICAL ADDRESS
23	2A	A3	08	0158 332 : BBS #IRPSV_PHYSIO,IRPSW_STS(R3), 10\$ ;BYPASS CONVERSION IF PHYSICAL I/O
				333 : MOVZBL (R2), RT : GET CURRENT LOGICAL SECTOR
				334 : CMPB #12, R1 : SET C IF 12 < SECTOR <= 26
				335 : ADWC R1, R1 : DOUBLE SECTOR NUMBER, ADD INTERLEAVE FACTOR
				336 : MOVZBL 2(R2), R0 : GET CYLINDER NUMBER
				337 : EMUL #6, R0, R1, R0 : COMPUTE SKEW (6 + CYL + SECT)
50	51	50	02	0166 338 : MOVZBL UCBSB_SECTORS(R5), -(SP) : GET SECTORS/TRACK
				339 : EDIV (SP)+, R0, R0, R1 : MODULO SECTOR INTO SECTORS PER TRACK
51	50	50	06	016A 340 : INCL R1 : OFFSET SECTOR NUMBER BY 1
				341 : MOVB R1, (R2) : SAVE SECTOR NUMBER
				342 : INCBL 2(R2) : INCREMENT PAST UNUSED CYLINDER
				343 : 10\$: :
			05	0180 344 : RSB :
				0181 345 :
				0181 346 : + XFER - TRANSFER DATA TO OR FROM USER
				0181 347
				0181 348 : THIS ROUTINE IS ENTERED VIA A BSB TO TRANSFER ONE SECTOR'S WORTH OF DATA
				0181 349 : TO OR FROM THE USER'S PROCESS.
				0181 350
				0181 351 : INPUTS:
				0181 352
				0181 353 : 0181 354 : R5 = ADDRESS OF UCB
				0181 355 : UCBSW_DX_BCR = BYTES LEFT TO TRANSFER
				0181 356
				0181 357 : OUTPUTS:
				0181 358 :

0181 359 : RO LSB SET = MORE DATA TO TRANSFER  
 0181 360 : UCB\$W DX BCR = COUNT OF BYTES REMAINING  
 0181 361 : RO LSB CLEAR = NO MORE DATA TO TRANSFER  
 0181 362 :  
 0181 363 : -  
 0181 364 :  
 0181 365 : .ENABL LSB  
 0181 366 :  
 0181 367 XFER:  
 00D8 C5 7E D4 0181 368 CLRL -(SP) : ASSUME REQUEST COMPLETE  
 08 68 A5 45 10 0183 369 BSB8 SETBUF : SETUP TRANSFER PARAMETERS  
 00D8 C5 52 A2 0185 370 SUBW R2,UCBSW DX BCR(R5) : UPDATE BYTE COUNT  
 02 13 018A 371 BBC #UCBSV\_DX\_WRITE,UCBSW\_DEVSTS(R5),10\$ ;BRANCH IF READ REQUEST  
 10 13 018F 372 BEQL 30\$ : IF EQL DONE  
 37 10 0191 373 BSB8 SETBUF : GET MORE DATA FROM USER  
 10 10 0193 374 BSB8 MOVFRUSER :  
 08 11 0195 375 BRB 20\$ : EXIT SUCCESSFULLY  
 00D8 C5 17 10 0197 376 10\$: BSBB MOVTouser : MOVE DATA TO USER  
 02 13 0199 377 TSTW UCB\$W DX BCR(R5) : TRANSFER COMPLETE NOW?  
 01A5 02 13 019D 378 BEQL 30\$ : IF EQL YES  
 00D8 C5 6E D6 019F 380 20\$: INCL (SP) : SET SUCCESS  
 01A5 01A1 381 30\$: MOVL (SP)+,R0 : SET SUCCESS  
 50 8E D0 01A1 382 05 01A4 383 RSB :  
 01A5 384 :  
 01A5 385 :  
 01A5 386 : .DSABL LSB  
 01A5 387 :  
 01A5 388 :+  
 01A5 389 : MOVFRUSER - MOVE DATA FROM USER TO FLOPPY BUFFER  
 01A5 390 :  
 01A5 391 : INPUTS:  
 01A5 392 :  
 01A5 393 : R1 = ADDRESS OF FLOPPY BUFFER  
 01A5 394 : R2 = BYTE COUNT  
 01A5 395 : R5 = ADDRESS OF UCB  
 01A5 396 :  
 01A5 397 : OUTPUTS:  
 01A5 398 :  
 01A5 399 : THE CONTENTS OF THE INTERNAL BUFFER ARE COPIED FROM THE USER'S  
 01A5 400 : ADDRESS SPACE.  
 01A5 401 :  
 01A5 402 :  
 01A5 403 : -  
 01A5 404 : .ENABL LSB  
 01A5 405 :  
 01A5 406 : MOVFRUSER:  
 00000000 54 52 D0 01A5 407 MOVL R2,R4 : SAVE BYTE COUNT  
 GF 09 11 01A8 408 JSB G^IOCSMOVFRUSER : MOVE DATA  
 01AE 409 BRB 10\$ : UPDATE BUFFER ADDRESS  
 01B0 410 :  
 01B0 411 :+  
 01B0 412 : MOVTOUSER - MOVE CONTENTS OF FLOPPY INTERNAL BUFFER TO USER  
 01B0 413 :  
 01B0 414 : THIS ROUTINE IS CALLED TO TRANSFER THE CONTENTS OF THE FLOPPY DATA  
 01B0 415 : BUFFER TO THE USER'S ADDRESS SPACE.

01B0 416 :  
 01B0 417 : INPUTS:  
 01B0 418 :  
 01B0 419 : R1 = ADDRESS OF FLOPPY BUFFER  
 01B0 420 : R2 = BYTE COUNT  
 01B0 421 : R5 = ADDRESS OF UCB  
 01B0 422 :  
 01B0 423 : OUTPUTS:  
 01B0 424 :  
 01B0 425 : THE FLOPPY BUFFER CONTENTS ARE COPIED TO THE USER'S ADDRESS SPACE  
 01B0 426 :  
 01B0 427 : -  
 01B0 428 :  
 01B0 429 MOVTouser:  
 00000000'GF 52 D0 01B0 430 MOVL R2,R4 : SAVE BYTE COUNT  
 7C A5 54 16 01B3 431 JSB G^IOCSMOVTouser : MOVE DATA TO USER'S BUFFER  
 7C A5 FE00 8F A0 01B9 432 10\$: ADDW R4,UCBSW\_BOFF(R5) : UPDATE PAGE OFFSET  
 04 04 12 01C3 433 BICW #^C<^X01FF>,UCBSW\_BOFF(R5) : MAKE MODULO PAGE SIZE  
 78 A5 04 C0 01C5 434 BNEQ 20\$ : IF NEQ PAGE BOUNDARY NOT CROSSED  
 05 01C9 435 ADDL #4,UCBSL\_SVAPTE(R5) : UPDATE ADDRESS OF USER PTE  
 01CA 436 RSB :  
 01CA 437 20\$: .DSABL LSB :  
 01CA 438 :  
 01CA 439 :  
 01CA 440 :  
 01CA 441 :  
 01CA 442 :+ SETBUF - SETUP PARAMETERS FOR TRANSFER TO OR FROM USER'S BUFFER  
 01CA 443 : THIS ROUTINE IS ENTERED VIA A BSB TO INITIALIZE ALL PARAMETERS REQUIRED  
 01CA 444 : TO TRANSFER ONE SECTOR OF DATA TO OR FROM THE USER'S PROCESS.  
 01CA 445 :  
 01CA 446 :  
 01CA 447 :  
 01CA 448 : INPUTS:  
 01CA 449 :  
 01CA 450 : R5 = ADDRESS OF UCB  
 01CA 451 :  
 01CA 452 : OUTPUTS:  
 01CA 453 :  
 01CA 454 : R1 = ADDRESS OF SECTOR BUFFER  
 01CA 455 : R2 = NUMBER OF BYTES TO TRANSFER TO OR FROM USER  
 01CA 456 : UCBS\$B\_SECTCNT = 128  
 01CA 457 : UCBSL\_DXBFPNT = ADDRESS OF SECTOR BUFFER  
 01CA 458 : -  
 01CA 459 :  
 01CA 460 SETBUF:  
 50 00D8 C5 3C 01CA 461 MOVZWL UCBSW\_DX\_BCR(R5),R0 : GET COUNT OF BYTES REMAINING  
 52 80 8F 9A 01CF 462 MOVZBL #128,R2 : ASSUME FULL SECTOR TO TRANSFER  
 00DA C5 52 90 01D3 463 MOVB R2,UCBS\$B\_DX\_SCTCNT(R5) : RESET SECTOR COUNT  
 51 00CC C5 00 01D8 464 MOVL UCBSL\_DX\_BUF(R5),R1 : GET BUFFER ADDRESS  
 00D0 C5 51 D0 01DD 465 MOVL R1,UCBSL\_DXBFPNT(R5) : SET ADDRESS  
 50 52 B1 01E2 466 CMPW R2,R0 : SECTOR EXCEED BYTES LEFT  
 03 03 1F 01E5 467 BLSSU 10\$ : IF LSSU NO  
 52 50 D0 01E7 468 MOVL R0,R2 : SET COUNT TO SMALLER AMOUNT  
 05 01EA 469 10\$: RSB :  
 01EB 470 :  
 01EB 471 :  
 01EB 472 DXP\_END:: :

DXUTILITY  
V04-000

- FLOPPY DISK DRIVER UTILITY ROUTINES  
START I/O OPERATION

L 12

01EB 473  
01EB 474 .END

15-SEP-1984 23:57:49 VAX/VMS Macro V04-00  
5-SEP-1984 00:14:19 [DRIVER.SRC]DXUTILITY.MAR;1 Page 11 (1)

ACPSACCESS	*****	X	02	UCBSL_DX_BFPNT	= 000000D0
ACP\$DEACCESS	*****	X	02	UCBSL_DX_BUF	= 000000CC
ACP\$MODIFY	*****	X	02	UCBSL_IRP	= 00000058
ACP\$MOUNT	*****	X	02	UCBSL_MEDIA	= 000000BC
ACP\$READBLK	*****	X	02	UCBSL_SVAPTE	= 00000078
ACP\$WRITEBLK	*****	X	02	UCBSM_CANCEL	= 00000008
DX\$ERR	0000011E	RG	02	UCBSM_INTTYPE	= 00000080
DX\$FUNCTIONAL	00000000	RG	02	UCBSM_POWER	= 00000020
DX\$STARTIO	00000070	RG	02	UCBSM_TIMEOUT	= 00000040
DXP-END	000001EB	RG	02	UCBSV_DX_WRITE	= 00000003
EXE\$SENSEMODE	*****	X	02	UCBSV_POWER	= 00000005
EXE\$SETCHAR	*****	X	02	UCBSW_BCNT	= 0000007E
FUNCTAB_LEN	= 00000070			UCBSW_BOFF	= 0000007C
FUNCTXT	00000131	R	02	UCBSW_CYLINDERS	= 00000046
IOSV_INHRETRY	= 0000000F			UCBSW_DEVSTS	= 00000068
IOS_ACCESS	= 00000032			UCBSW_DX_BCR	= 000000D8
IOS_ACPCONTROL	= 00000038			UCBSW_STS	= 00000064
IOS_CREATE	= 00000033			XFER	0000018, R 02
IOS_DEACCESS	= 00000034				
IOS_DELETE	= 00000035				
IOS MODIFY	= 00000036				
IOS_MOUNT	= 00000039				
IOS_READLBLK	= 00000021				
IOS_READPBLK	= 0000000C				
IOS_READVBLK	= 00000031				
IOS_SENSECHAR	= 0000001B				
IOS_SENSEMODE	= 00000027				
IOS_SETCHAR	= 0000001A				
IOS_SETMODE	= 00000023				
IOS_VIRTUAL	= 0000003F				
IOS_WRITELBLK	= 00000020				
IOS_WRITEPBLK	= 00000008				
IOS_WRITEVBLK	= 00000030				
IOC\$DIAGBUFILL	*****	X	02		
IOC\$MOVFRUSER	*****	X	02		
IOC\$MOVTouser	*****	X	02		
IOC\$REQCOM	*****	X	02		
IOSUCC	0000012E	R	02		
IRPSL_MEDIA	= 00000038				
IRPS\$FCODE	= 00000006				
IRPSV\$FCODE	= 00000000				
IRPSV\$PHYSIO	= 00000008				
IRPSW\$FUNC	= 00000020				
IRPSW\$STS	= 0000002A				
MASKH	= 00000008				
MASKL	= 04000000				
MOVFRUSER	000001A5	R	02		
MOVTouser	000001B0	R	02		
SETBUF	000001CA	R	02		
SSS_IVADDR	= 00000134				
SSS_NORMAL	= 00000001				
TRK\$SEC	0000014B	R	02		
UCBSB\$DX_SCTCNT	= 000000DA				
UCBSB\$ERTCNT	= 00000080				
UCBSB\$ERTMAX	= 00000081				
UCBSB\$SECTORS	= 00000044				
UCBSL\$DEVDEPEND	= 00000044				



0111 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

